UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/675,117	09/30/2003	Anuja Deedwaniya	POU920030128US1	6058

7590 12/07/2007

Philmore H. Colburn II
Cantor Colburn
55 Griffin Road South
Bloomfield, CT 06002

EXAMINER	
CHOW, CHIH CHING	
ART UNIT	**PAPER NUMBER**
2191	

MAIL DATE	DELIVERY MODE
12/07/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/675,117	DEEDWANIYA ET AL.	
	Examiner	Art Unit	
	Chih-Ching Chow	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 November 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-9 and 11* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-9, 11* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *24 October 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☒ Interview Summary (PTO-413) Paper No(s)/Mail Date. *10/11/07*.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to amendment dated September 04, 2007.

2. Per Applicants' request, no claim is amended.

3. Claims 1-9 and 11 remain pending.

4. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on November 07, 2007 has been entered.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claims 1, 9 and 11 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claims 1, 9, and 11 recite "compiling and linking said source file iteratively to create a single executable file" – the 'compiling and linking said source file iteratively' feature is only mentioned in the Abstract, it's not clear to the Examiner what is the reason to do the compiling and linking 'iteratively' in order to create a single executable file? Is that meant to do compiling and linking and create a single executable for each of the variations, characteristics, and parameters for each attribute? If so, what is the naming

convention for each of the executables (assuming they should all be different)? The feature of the 'iterative' feature is not clearly described in the Specification.

7. Claims 2-8 depend on claim 1, they are rejected under 35 USC § 112 (2) for the same reason.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 1-6, 8, 9, and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over US No. 6,442,752, by Jennings et al., hereinafter "Jennings"; in view of U.S. Patent No. 6,295,642 by Blandy, hereinafter "Blandy".

As Per claim 1,

- ***A method of packaging a dynamically linked computer program function comprising:***

- ***establishing an attribute, each attribute exhibiting a plurality of at least one of variations, characteristics and parameters associated with said dynamically linked computer program function;***

The concept of establishing an attribute, with a plurality of definitions via various parameters, is to 'overload a method' in a computer program, which is a well known skill to the people in the art at the time of the invention was made.

- ***obtaining a source file associated with said dynamically linked computer program function;***

Jennings's disclosure is a method of a dynamically linked computer program, see Jennings's 'Description of the Prior Art', column 1, lines 35-38, "Most computer operating systems provide a **dynamic link library facility** that enables a computer programmer to provide certain **executable procedures and functions** in the form of **a library** that is **separate from the applications** (*obtaining a source file*) that execute on the computer." -- dynamic linking of computer program functions is a well-known skill to the people in the art. Jennings's disclosure also teaches establishing an attribute with said associated dynamically linked computer program function, see Jennings's Fig. 4, and description in column 3, lines 3-28, "the template 24' includes a Header block 34, an Imports block 36, a Types block 38, a Names block 40, and an **Attributes** block 42. The Header block 34 appears physically at the beginning of the template 24' and contains, in fixed locations, pointers to the other sub components. The Names block 40 contains the names (character strings) of the items that the application program has declared as imports. The Types block 38 contains encoded descriptions of the types of the imported procedures and their **parameters**. The imports block 36 contains an entry for each imported item. Each entry in the imports block 36 contains a pointer to the name (within the Names block 40), and the type signature (within the Types block 38) of the imported item. As in the directory 22', the type signatures are in the form of a recursive description, with the kind of item (procedure, function, etc.) at the top. This is followed by the return value type, if any, and then the

parameters in order. If the **parameters** have additional structure (such as an array), that information is included within the description for the **parameter**. Each element of a type description carries a length indication which includes any subordinate information. Finally, the **Attributes** block 42 contains information about the **attributes** of the requested library. One series of **attributes,** for example, specifies the file name of the library code file to which the application will **dynamically link."** -- attribute exhibiting via parameter(s).

- ***compiling and linking said source file iteratively to create a single executable file based on said at least one of variations, characteristics, and parameters for each said attribute; and wherein said single executable file is configured to facilitate choice of a selected version of said function based on a particular said at least one of variations, characteristics, and parameters for each said attribute.***

For dynamic compiling and linking feature see Jennings's Figure 4, a graphical depiction of a template that is generated during **compilation of an application program** that calls one or more of the procedures exported by the **dynamic link library** of FIG. 2 (*compiling and linking*). For creating an executable file based on given parameters feature, see Jennings's column 10, lines 10-21, "Lines 909-924 comprise a set of inline functions that provide array **bounding code to support passing parameters** of data type INTEGER ARRAY to the exported procedure UNTYPED_PROCEDURE (see FIG. 2). Lines 925-934 comprise two inline functions that **support passing arguments by reference**. These functions are used in the present example to **pass parameters of data type** REAL_PARAM by

REFERENCE to the exported procedure UNTYPED_PROCEDURE. Lines 935-952 comprise two inline functions that perform externally signed octal-to-twos complement conversion on parameters of data type INTEGER (and vice versa)." Jennings teaches all aspects of claim 1 but he does not mention 'compiling and linking iteratively' specifically, however, Blandy teaches it in an analogous prior art, see Blandy's Abstract, "**An iterative process is employed whereby bytecodes are compiled up to the next conditional flow bytecode** or return, the compiled code is executed and any attempt to enter uncompiled paths of the method is monitored. When the executing thread attempts to execute an uncompiled path control is returned to the compiler and **the process is repeated** starting with the first bytecode of that path."

It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to supplement Jennings's disclosure of the method of dynamically linking of overloading attributes with iterative compilation, taught by Blandy. The modification would be obvious because one of ordinary skill in the art would be motivated to iteratively compile all paths of the bytecode (column 6, lines 44-48).

As Per claim 2,

- ***The method of claim 1 further including configuring an application to be responsive to said selected version of said function based on said particular at least one of variations, characteristics, and parameters for each said attribute.***

For claim 1 feature see claim 1 rejection, for the 'configuring' feature see Jennings's column 3, lines 32-35, "The description of the imported procedure in the **library template** (*configuring an application*) of the calling application is then compared to the description of the exported procedure in the directory of the library code file to ensure that the parameters match." – the parameters matching process ensures that the selected version of said function based on a particular parameters is in use.

As Per claim 3,

- ***The method of claim 2 wherein said configuring includes compiling based on said particular at least one of variations, characteristics, and parameters for each said attribute.***

Claim 2 rejection is incorporated, for rest of claim 3 feature see claim 1 and 2 rejections.

As Per claim 4,

- ***The method of claim 1 wherein said attribute includes: version, addressability, character code base, character set that a specific operating system supports, system linkage conventions; machine architecture, or floating point hardware.***

Claim 2 rejection is incorporated, for rest of claim 4 feature see Jennings's column 9, lines 49-54, "FIGS. 9A-9C, 10, and 11A-11B are source code listings of an exemplary global header file, prototype header file, and **Win32DLL** (for Microsoft Windows NT operating system, see Jennings's column 3, lines 46-47) **source code** (*a specific operating system*

supports) skeleton file, respectively, generated by code generator 104 in accordance with the method of the present invention using a **compiled version** of the MCP-based dynamic link library listed in FIG. 2" and column 9, lines 60-63, "Referring to FIGS. 9A-9C, the global header file (globals.h) contains global definitions and the C++ classes required to perform the **parameter mapping functionality** (*attribute*) of the interface jacket routines." And column 10, lines 18-21, "Lines 935-952 comprise two inline functions that perform externally signed **octal-to-twos complement conversion** on parameters of **data type INTEGER** (and vice versa) – *machine architecture*".

As Per claim 5,

- ***The method of claim 1 wherein said attribute is user specified.***

For claim 1 feature see claim 1 rejection, for rest of claim 5 feature see Jennings's column 1, lines 35-38, "Most computer operating systems provide a **dynamic link** library facility that enables a **computer programmer** to provide certain executable procedures and functions in the **form of a library** that is separate from the applications that execute on the computer." And column 8, lines 59-62, "Specifically, the **name of each exported procedure** of the first DLL is extracted, along with information identifying the **number, order, and data type of each parameter** (i.e., argument) that can be passed to the procedure."– the parameters are user specified attributes for a program.

As Per claim 6,

- ***The method of claim 1 wherein said attribute is implicitly defined.***

For claim 1 feature see claim 1 rejection, for rest of claim 6 feature see Jennings's column 13, lines 47-59, "These new attributes can be set for a calling application in the MCP environment at compile time, either explicitly through programming language constructs, **or by default in the absence of such constructs** *(implicitly)*. However, in order to achieve transparency of the library replacement mechanism of the present invention, a 'library equation' facility of the MCP environment is employed. This facility **allows attribute values established for a given application by its compiler to be overridden at run time**, either through MCP work-flow language (WFL) constructs associated with the initiation (run) of the application, or corresponding programming language constructs (task attribute assignments) when the application is initiated directly from some other program."

As Per claim 8,

- ***The method of claim 1 wherein said at least one of variations, characteristics, and parameters for each said attribute is user specified.***

For claim 1 feature see claim 1 rejection, for rest of claim 8 feature see claim 5 rejection.

As Per claim 9,

- ***A storage medium encoded with a machine-readable computer program code, said code including instructions for causing a computer to implement a method of packaging dynamically linked computer program function, the method comprising: establishing an attribute, each attribute exhibiting a plurality of at least one of variations, characteristics and***

parameters associated with said dynamically linked computer program function; obtaining a source file associated with said dynamically linked computer program function; compiling and linking said source file iteractively to create a single executable file based on said at least one of variations, characteristics, and parameters for each said attribute; and wherein said single executable file is configured to facilitate choice of a selected version of said function based on a particular said at least one of variations, characteristics, and parameters for each said attribute.

Claim 9 is a computer implemented method for a plurality of software applications version of claim 1, it is rejected on the same basis as claim 1.

As Per claim 11,

- ***A system for packaging a dynamically linked computer program function comprising: a means for establishing an attribute, each attribute exhibiting a plurality of at least one of variations; characteristics and parameters associated with said dynamically linked computer program function; a means for obtaining a source file associated with said dynamically linked computer program function; a means for compiling and linking said source file iteratively to create a single executable file based on said at least one of variations, characteristics, and parameters for each said attribute; and wherein said single executable file is configured to facilitate choice of a selected version of said function based on a particular said at least one of variations, characteristics, and parameters for each said attribute.***

Claim 11 is a system implemented version for a dynamically linked computer program function of claim 1, it is rejected on the same basis as claim 1.

10. Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over US No. 6,442,752, by Jennings et al., hereinafter "Jennings"; in view of U.S. Patent No. 6,295,642 by Blandy, hereinafter "Blandy"; further in view of U.S. Patent No. 6,735,598 by Srivastava (hereinafter "Srivastava").

11. As Per claim 7,

- ***The method of claim 1 wherein said at least one of variations, characteristics, and parameters for each said attribute includes, 64-bit versus 32-bit addressing; ASCII versus EBCDIC, internal representation for character data, or HEX versus IEEE representation to use for floating point data.***

Jennings and Blandy teach all aspects of claim 7, but he does not disclose '64-bit versus 32-bit addressing; ASCII versus EBCDIC, internal representation for character data, or HEX versus IEEE representation to use for floating point data' explicitly, however, Srivastava teaches 'ASCII versus EBCDIC' in an analogous prior art; see Srivastava's Figure 12 and 13, and description in column 15, lines. 8-13, "A problem with digital representations generally is that there may be different digital representations of the same thing. For example, in some computer systems, characters are represented using **ASCII character codes; in others, they are represented using EBCDIC codes.** In both cases, what is represented

is characters, and the same operations are performed with characters using either representation; what is different is the format used to represent the characters." And column 15, lines 29-31, "The solution provided in the database management system described herein is to **define a FORMAT package for each of the different formats.** The package contains methods for reading each of the formats and a method that permits general processing of the format, including translation form one format to another." Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention was made to supplement Jennings's and Blandy's disclosure of the method of dynamically linking of various attributes defined by parameters and overloading functions with various parameter types by using format conversions for ***ASCII versus EBCDIC***, or ***64-bit versus 32-bit addressing*** etc., taught by Srivastava. The modification would be obvious because one of ordinary skill in the art would be motivated to include a **format attribute** which specifies a particular format package for different application programs (Srivastava column 4, lines 44-50).

Conclusion

12. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Chen et al., US 5,930,795, discloses a framework for a query compiler and run-time environment for resolving a table reference to a dynamic table that is first identified at run-time but is initially unknown at compile-time. A parser parses the table reference and creates a parsed representation for the table that identifies the type of dynamic table. A code generator creates executable plans

containing run-time table object representations (TAOB), from the parsed representations, that contain the type of dynamic table. The TAOB is also extended to provide for parameters that are definable at run-time, including a table ID of the actual table entity being referenced.

Baseline, "Use method overloading in JAVA", discloses an example of reusing method names via overloading.

13. The following summarizes the status of the claims:

35 USC § 102 rejection: Claims 1-6, 8, 9, and 11

35 USC § 103 rejection: Claims 7

35 USC § 112 (2) rejection: Claims 1-9, and 11

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chih-Ching Chow whose telephone number is 571-272-3693. The examiner can normally be reached on 8:30am - 5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Any inquiry of a general nature of relating to the status of this application should be directed to the **TC2100 Group receptionist: 571-272-2100**.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-

direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Chih-Ching Chow
Examiner
Art Unit 2191
November 29, 2007

CC

MARY STEELMAN
PRIMARY EXAMINER